A Methodology for Quantitative Estimates for the Work and Disturbance Transformation Matrices*

P C Matthews† and **C D W Lomas**
School of Engineering, Durham University
Durham DH1 3LE, UK

July 20, 2010

Abstract

Modern design projects are typically undertaken concurrently in a virtual enterprise network of expert design and manufacture agents. The general need for agile response in turbulent environments is well documented and has been analysed at the manufacture phase. This paper proposes a framework to enable the simulation and analysis of an agile design methodology. This framework models the occurrence of an unexpected local event in a concurrent design project and how it propagates to the global project. The redistribution of the design work can be controlled within the virtual enterprise and the total redistribution impact can be measured. A four level classification scheme for the severity of unexpected events is proposed. A trial design experiment is conducted, and a first order quantitative analysis is performed based on Work Transformation Matrices (WTM) and a novel Disturbance Transformation Matrix (DTM). A design negotiation process based on the WTM/DTM is proposed.

1 Introduction

The concepts of Agility and Design are well defined. Agility is seen to be the ability to rapidly react to changes in the environment, be they expected or not. When applied to manufacture, this is seen to be the ability to change a manufacture resource rapidly to produce a different assembly or product. Thus, an agile manufacturer is able to rapidly respond to changes in product demand. Design is the process of transforming a set of potentially ill-defined customer requirements into an implementable solution. This process is undertaken for the design of physical products, systems, processes, and other services.

*Journal of Engineering Design, Volume 21, Issue 4 August 2010, pages 413–425
†email: p.c.matthews@durham.ac.uk

The design process is a combination of determining appropriate attributes for a product or system and then determining their values and/or configuration. A number of methodologies exist for deployment in different application areas and different corporate cultures. As designs have become more complex, it has been necessary to migrate to concurrent design methods [1]. However, concurrent design methods lack the ability to respond to unpredicted changes in environmental conditions.

Agile Design is therefore a design process in which the design team is able to rapidly respond to a change in the design specification. These changes can either arise from a customer providing a revised specification or from the design process itself, for example when different aspects of the design introduce a conflict that must be resolved. It is assumed that these events are unexpected, and therefore not planned for. Agile Design seeks to mitigate the impact of these events on the total design process.

The need for diverse design teams is a shared property between Agile Design and Concurrent Design. These are both instances of Virtual Enterprises as they both lever resource networks to solve a design problem [2]. The distinction between these two methodologies is the greater responsiveness that is core to Agile Design. Given a design brief, both methods use competence profiling to suitably allocate design tasks to a pool of design agents. When an unpredicted event occurs, which could be either a late customer request, the failure of an agent, or some other external environmental impact, the concurrent design process is interrupted. The impact of the event is evaluated, and the design process is restructured accordingly but with minimal impact to unaffected agents. The aim is to minimise the total impact to the design process. However, at the same time it is important to minimise communication between active agents as this delays progress. This is achieved by not only reviewing the directly impacted agents, but also their interfacing agents and their associated latencies. These two requirements are in conflict as to minimise total impact on the design requires thorough discussion between all relevant designers, which in turn requires significant time to be taken away from the total design process. These two contradicting requirements provide the basis for optimising the management of the concurrent design process, thus providing the agility.

This paper will develop a set of foundations that will characterise Agile Design and provide a set of requirements that must be met to enable Agile Design. A review of general Agile Design scenarios is presented, along guidelines for appropriate responses to ensure agile reaction. A model for project completion rate is presented and augmented to take into account unexpected disturbances. A laboratory based design trial is then presented, which is analysed to estimate numerical values for the local (individual) work rates and the quantitative impact that an unexpected event has on the overall design progress. This is then used as a basis for illustrating how the numerical model can be used to direct the designers to optimally undertake design negotiation once an unexpected event occurs.

2 Background

At its most basic, the design process takes us from an initial requirements specification through to the final set of manufacture plans [3]. This is represented by a single work thread, marked by a set of check points representing the design moving from one development phase to the next. It is at these check points where unsatisfactory work can be referred back, either to the start of that phase or further back if more significant errors are found. These design phases can be seen to consume time as a primary resource. However, this simple representation does not provide for concurrency of work. As such, it forms the terminal part of a concurrent engineering project.

Concurrent engineering is the distribution of the design, and potentially manufacture, work between a number of agents [4]. These agents can then either recursively apply concurrent engineering again, or follow the basic linear design process if they are a 'terminal' agent. The agents are selected according to their known expertise [5]. These agents are networked through a virtual enterprise while the project is underway. Through this network, agents communicate as necessary. This concurrent engineering approach provides a means for rapidly creating enterprises with high degrees of competency without the need to support these competencies during projects which do not require the same competency profile. Thus, the virtual enterprise has the benefits of a large, well found enterprise, without having to pay for the maintenance overhead of resources that are not required for other projects.

The concept of agility in a manufacturing context has recently emerged [6, 7]. Most authors agree that 'agility' is the ability to rapidly respond to some external and unexpected event. The argument promoting agility is that it enables better survival in turbulent market conditions. However, most agile responses are tailored to changes in product demand, either in the form of production levels or in alternate design. The solutions to these tend to fall in line with traditional manufacture theory (for example, by applying Just-in-Time methods) or design modification (such as mass customisation applied after the initial product launch). Thus, enterprises use the agile methods to enable them to respond to the market, based on a given design.

Where designs require modification, change management methods provide a structure to enable managed modification of a design or manufacture process [7, 8]. There are two aspects of change management: (1) the process of changing a given design, and (2) the design of artefacts such that they are more easily modified when needed. In the process of changing a given design, care must be taken to avoid a change in one part of the design negatively impacting some other part. This requires not only an understanding of how all aspects of the design interact with each other, but also a mechanism to rectify such collisions when they occur. This can be particularly troublesome when the two conflicting design aspects are 'owned' by different agents in a virtual enterprise. If change is expected to a design, then the original design can be made such that changes can be easily applied. This idea has also been extended to unexpected change, but where the nature of the change has been anticipated, and therefore either a case-based reasoning approach can be adopted [9, 10] or a set of change procedures can be prescribed [11, 12]. Beyond this, there

is relatively little literature on changing the design mid-way through the design process. This tends to be subsumed under failing a mid-way design check point, thereby sending the design back to an earlier phase. However, such changes arise frequently due to the customer modifying the original specification.

The Design Structure Matrix (DSM) provides a means for representing the causal information exchange structure between tasks within a design project [13]. This approach has re-emerged as an active research area, in the first instance due to the ability of the DSM to represent iterations within the design process [14, 15, 16, 17, 18]. Iterations arise where two tasks mutually require information from each other: working on one task will induce a small change in the other task, and vice-versa. The amount of rework per cycle for a set of fully concurrent tasks has been quantified using the Work Transformation Matrix (WTM) [14]. This provides a means for identifying the tasks that are most likely to require a greater number of iteration before arriving at an acceptable solution. The basic WTM approach assumes a static design scenario. Further work has considered the dynamic nature of the product development process, and allowing for small perturbations [17]. Another approach considers the structure of the WTM to identify how to optimally deploy resources to the various tasks to improve the convergence rates of the various task completion levels [18]. Both these extensions primarily concern the resource deployment to the design process. Two alternative perspectives on this analysis is to consider where the resource availability changes and the unexpected (or externally induced) rework imposed on a task or set of tasks.

Finally, it is important to consider the human aspects of engineering projects. Where multiple agents are responsible for different but interacting design aspects, it is important that all parties co-operate. However, evidence shows that human nature tends to 'hide' problems in the hope that they can be resolved without needing to admit there ever was a problem [19]. When the problem is not resolved, it then has an amplified effect on the remainder of the design when it can no longer be hidden. Frequently, resolving these problems requires changes to the design which results in the design process moving backwards. While an agile design process will not change the problems arising due to human factors, it can mitigate the effects of these factors when they arise.

This paper will address how the design process can be made agile. By using concurrent engineering methodology as a starting point, a framework will be proposed that will enable analysis of how unexpected events affect the concurrent design process. Given this analysis tool, it then becomes possible to test different design scenarios to handle an unexpected event occurring during the design process. This framework will be strictly concerned with the temporal aspects of a design project, and not with any other resource use. However, the underlying principle can be readily adopted to these different resources.

3 Analysis Framework

The basis of this agile design analysis framework stems from concurrent engineering and virtual enterprise methodology. Specifically, given a design project, the project is decom-

posed into a number of independent tasks that are undertaken by different design agents. The concurrent engineering view assumes that the next global phase of the project will then be the combining of the component solutions into the final deliverable.

The analysis framework is based on the WTM to provide a quantitative model of the design process [14]. Using the WTM, the level of work remaining on each task is represented as a vector, $\mathbf{x}$. With the linear model, the work remaining on each task after iteration t is given by:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) \tag{1}$$

It should be noted that this only represents a set of concurrent tasks, as opposed to iterations within a set of sequential tasks. However, for the purposes of this paper, it is sufficient to restrict the framework to the pure concurrent case.

The agile design framework considers what happens when some interruption occurs to a design agent. The interruption events to be considered are those that incur additional work to a task. This can be represented by adding a 'disturbance' term to Equation 1:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\delta(t) \tag{2}$$

where $\delta(t_e) \neq \mathbf{0}$, when the unexpected event occurs and $\mathbf{B}$ represents the 'disturbance transformation matrix' (DTM). The DTM represents how a significant disturbance in the project propogates through to the whole project. This model does not address communication latency: it is assumed that with modern communication technologies, this latency is not significant. It must be noted that there exist cases where this latency is significant, for example where consideration must be given as a result of a message. However, for the purposes of this analysis, such latencies will be beyond the scope of this study.

This paper will consider the case where only one task, i.e. one component of $\mathbf{x}$, is affected. However, the analysis can be expanded to consider events that would induce re-work in several tasks simulataneously. Under the single task analysis, the other tasks receive the additional work in the next cycle as a result of the WTM.

Depending on the nature of the event and how much slack was built into the project, such events potentially result in a global effect to the whole design project. For analysis purposes, four levels of (local) event severity have been categorised and these categories are described in terms of δ [20, 21]:

Trivial: the problem can be resolved completely at the local level, a small time penalty is incurred. $\mathbf{B}\delta$ is near-zero.

Minor: the problem requires the agent to seek external assistance, or minor redeployment of part of the work to another partner within the virtual enterprise. δ only has one non-zero component and $\mathbf{B}\delta$ is moderate.

Major: the problem cannot be resolved by the agent or other member of the virtual enterprise. A new member is needed to join the virtual enterprise, and the redeployment

of work and initiation of the new member to the project incurs a serious time penalty. δ has several non-zero components and $\mathbf{B}\delta$ is significant.

Fatal: the problem cannot be resolved by the agent, and there exists no external agent that can provide support. Effectively, the design is fundamentally flawed and is not realisable. δ is such that $\mathbf{A}\delta$, the resulting rework vector, is very close to $\mathbf{x}(0)$, the original level of work.

An event occurring during the design process effectively requires more time to be spent than originally planned. This can be simply represented by in effect 'turning the clock back' on the design process. By mapping the proportion of work remaining onto the design lifecycle, the interruption is encoded as being an event that requires a particular type of redesign work to be performed. The above categorical labels provide a fuzzy-like mapping between an unexpected event and the quantitative δ value that is to be added. This provides an initial estimate to compute the amount of rework that will be induced into other tasks. So if an event happens in the detailing stage, the severity of the event determines how far back the whole project is set back. For example a trivial event in the detailing stage might only require the affected agent to restart this stage. The more serious 'minor' event might result in the agent returning to the embodiment stage while a 'major' event will result in a new agent starting this part of the design work from fresh. Another possibility in the 'minor'/'major' cases is that several other design agents are affected, to the respective degrees of the event classification. Finally, in the case of a 'fatal' event, it is assumed that the whole design collapses due to the event. In this case the design requires fundamental rework and hence all agents will start from fresh, effectively under a new virtual enterprise.

3.1 Event sources

External and unexpected events are the source of interrupts to the design process [22]. This paper shall distinguish three major sources of interrupt events: (1) late customer design brief change, (2) in-production design change, and (3) agent failure. The severity of the event will depend on the nature of the event and the ability of the relevant agent(s) to handle the event.

Customers, in general, do not provide flawless documentation and, depending on circumstances, request unexpected design changes before the end of project. This customer driven change requires the design to be reconsidered. At the trivial level, it could be a simple matter of requesting a different material for the external body. More significant requests are those that change the functionality or core design of the artefact.

In-production design changes are those that arise either internally due to identifying additional design constraints or novel design solutions. Finally, agent failure represents the event where an agent is not capable of delivering the solution as originally specified. The failure of an agent effectively causes a design change request, which needs to be suitably propagated through the virtual enterprise.

3.2 Concurrent Design Representation

By distributing the design tasks to independent agents, each agent can be seen to perform a classical single threaded design project. Using the Pahl and Beitz design phase categories [3], the normal state progression of any agent would be: Conceptual, Embodiment, Detail, and Manufacture. Depending on the design task and the agent's expertise, each agent will require a different amount of time to complete each stage. The sum of these times represents the total time an agent requires to complete its task, assume no interruptions. Note that using this scheme does allow for agents to specialise in particular aspects of the design process, for example a manufacturing agent will require zero time for the prior design activities.

3.3 Agile Design Response

The state of the whole virtual enterprise is given by $\mathbf{x}$, the work remaining per agent vector. When an interrupting event occurs to an agent, the effect is propagated through the WTM which represents the causal links of the global design project within the virtual enterprise. The two extreme event types (trivial and fatal) do not require an agile design methodology to mitigate as they either are absorbed locally or globally destroy the project. However, the minor and major event classes do provide an opportunity to apply agile design methods to minimise the *total* time penalty to the project.

To respond with agility requires that when an event occurs in the concurrent design process, all design agents are able to respond accordingly. In a non-agile system, the penalty is accumulated totally by the affected agent. In the agile system, other agents aim to reduce the impact of the event by being able to self-incur trivial events. The unmitigated impact of the event can be identified by computing $\mathbf{B}\delta(t_e)$. This vector represents how the additional work would be transfered to the other agents after one cycle.

Thus the required change due to the source event is in effect mitigated through the relevant partners in the virtual enterprise. The aim of this analysis framework is to model and simulate such events to be able to test mitigation policies under laboratory conditions.

4 Trial Experiment

To investigate the potential of the agile design framework, a trial experiment was run. The experiment was based on designing a simple spoked bicycle wheel. The design process was recorded, and subsequently analysed to numerically estimate the WTM and the DTM. Finally, the impact of the 'unexpected' event was assessed.

The primary aim of the trial experiment was to test the interupt method and subsequent design negotiation. This represents the main agile design tool, namely the redistribution of the design rework between the design agents. For this trial, all the design and negotiation work was left unsupervised. It was up to the designers themselves to determine how best to redistribute the work load. The observation of this process represented the secondary aim of the trial: to identify what information would designers need to optimise the negotiation

process. The aims of this experiment are to investigate the viability of collecting design event data and to verify that the DTM is a reasonable process model, one trial will suffice.

The trial was performed using three researchers from within the School of Engineering. They were all proficient, but not expert, at using the SolidWorks CAD package.

4.1 Design Problem

The design task was to use a commerical CAD package to produce the detailed design for a spoked bicycle wheel. The design was partitioned into three components: the wheel rim, the spokes, and the wheel hub. The designers were supplied with sketches of each component and a basic specification of the wheel (rim width, depth, and outer diameter; and the inner hub diameter). Each designer was randomly allocated one component to design. With each component, further domain knowledge was supplied. For each component, this contained a series of suggested alternative designs and a rough indication of how difficult each would be to achieve. This was to facilitate any design negotiation that might occur.

The first phase of the design process was for the designers to agree on the overall shape of the wheel. This was relatively superficial (e.g. it was not required to consider the stress the wheel would undergo), and was used to identify further design parameters that would need to be agreed upon. These parameters determined the interfaces between the components, for example the spoke diameter determined the minimum size for the holes in the hub and rim.

Once the designers felt that they had sufficiently defined the problem, they entered the second phase of the design process. This was the detailed design phase in which the designers worked independently and represented the independent concurrent phase of the design project. On completing this phase, the components were checked to see that would fit together correctly, thus completing the design project.

4.2 Experiment set up and execution

The design experiment was based around the design of a simplified bicycle wheel. This provided a product with sufficient complexity (three interfacing components), but would be sufficiently familiar to the designers that it would not require significant introduction beyond supplying the designers with a small amount of 'specialist' knowledge that they could use as a basis for design negotiation.

The experiment was run over a 90 minute period. The first fifteen minutes were used to explain the objectives of the experiment and describe the data that was to be collected from the design process. During this briefing period, the 'agile interupt' method was described to the subjects. The subjects were to use the interupt method whenever they realised that their component would not interface easily with another. The interupt then started a design negotiation process which would aim to minimise the total time required to design the interface on both components. The subjects were also informed that at some point during the experiment, one designer would have a significant change imposed and that they would therefore be using the interupt method at some point.

The three designers were seated adjacently, in the order that the components interfaced to each other (hub–spoke–rim), and were randomly allocated a component to design. A video camera was used to capture the physical interactions between the designers and two observers took notes on the nature of the interactions. The designers were asked to save snapshots of their CAD status at regular intervals so that the progress could be tracked. In addition, the designers were asked to record whenever they requested information from or interupted another designer. This would then be correlated against the video record.

As stated in the design problem, the first stage of the design required the designers to determine and agree on the internal design parameters. The three designers spent about six minutes discussing the general arrangement of the wheel (basic geometry, number of spokes, component type selection). After this, the designers focused on their own components with one interuption to check on the spoke diameter.

The observers waited until the spoke design was about 50% complete before introducing the 'unexpected' event. The spoke design at that point was a thin rod, with a 90 degree bend at the base where it would connect into the hub. The event prohibited this bend in the spoke, and hence required the spoke to be redesigned. The spoke designer considered the possible alternatives, and decided to design a thicker spoke that would be screwed into the hub. As the spoke would be thicker, fewer would be required. Once this strategy had been decided by the spoke designer, the other two designers were interupted. The rim designer needed only to modify the hole geometry and the number of holes in the rim. The hub designer entered into negotiation with the spoke designer on the nature of the new hub–spoke interface. As a result of this negotiation, the new hub and spoke design were significantly different and basically needed to start from fresh. This event did not require an external member and the δ value only had one non-zero component (see Equation 5), hence the event is classified as a 'minor' event.

Once the new design had been negotiated between the spoke and hub designers and confirmed the final geometry and spoke numbers with the rim designer, both returned to work independently and completed their designs. Upon completion of all designers, a quick check was made to ensure the components would fit together successfully. Finally, a fifteen minute debrief was conducted.

4.3 Data Analysis

Data was collected from three main sources: the video recording of the trial; notes taken by the observers during the trial; and the frequent CAD snapshots taken. Further data on the amount of rework the participants felt they had to do was also collected during the debriefing session, however this was largely subjective and for comparison purposes only.

The quantitative measure for task completion was based on the number of remaining open problems. Each design task was analysed and all the open problems were enumerated (see Table 1). The open problems were classified into two types: (1) team based problems that needed negotiation and (2) implementation problems that were solved individually. The team based problems primarily resolved component interface issues and included identifying design parameters and their values. Implementation problems resolved

Table 1: Open problems per design task.

Task	Team problems	Individual problems	Total
Rim	15	9	24
Spoke	19	8	27
Hub	30	15	45

how to embody the chosen design in the CAD tool.

While it was not possible to identify when each individual open problem was solved (or re-opened), the trial data provided sufficient cues to identify when blocks of problems were solved. For example, the video record clearly shows when the rim designer leaves the discussion. At this point, it can be inferred that all designers have resolved all the team based problems that involved the rim. Similarly, when the remaining two designers break away to work individually, it is clear that the remaining team based problems have been cleared. The timestamped CAD snapshots provide data on the resolution of individual problems. Immediatelty after the unexpected event, detailed notes were taken by the observers with regard to the problems that were re-opened. Again, it was not always possible to identify individual problems, rather blocks of problems that were re-opened together. Again, the video record provides timed cues as to when these problems are opened and the subsequent negotiations. Table 2 contains the event log of the trial which is graphically represented in Figure 1.

4.4 Estimating the WTM and DTM

For the purposes of this trial, it was not appropriate to attempt to compute accurately the numerical values for the WTM and DTM as discussed in Section 3. However, it is interesting to consider the coarse first order estimates of the values of the matrices, as this provides insights into how the negotiations between the designers should be structured. A crude estimate of the initial disturbance can be read from the events log in Table 2. The disturbance event can be seen in the first instance in Spoke event number 5 ($t_e = 930$) where 12 tasks are re-opened. In the next step ($t_e + \varepsilon = 1012$), this event has propagated to the Hub, where 31 tasks are opened (and another 4 tasks are opened in the Spoke). It requires one more step before the event propagates to the Rim. As such, it can be inferred that the discrete timed WTM and DTM take the following form:

$$\mathbf{A} \approx \begin{pmatrix} 0.9986 & 0 & 0 \\ 0 & 0.9988 & 0 \\ 0 & 0 & 0.9978 \end{pmatrix} \quad \mathbf{B} \approx \begin{pmatrix} 1 & 2 & 0 \\ 0.5 & 1 & 4 \\ 0 & 0.25 & 1 \end{pmatrix}. \tag{3}$$

First, it is worth noting that the $\mathbf{A}$ matrix is diagonal. This represents that in this trial, provided there were no interruptions, the three components were able to be designed independently. This does represent an artificial case and is a direct result of (1) the

Table 2: Progression of the design for each sub-task.

Task	Id	t (s)	ΔO_{Task}	O_{Task}	Description
Rim	1	0	—	24	Start
	2	238	9	15	Team work ends
	3	560	12	3	Individual CAD work progressing
	4	1012	1	2	Progress slows due to awareness of spoke event
	5	1145	–4	6	Rim interupted by Hub and Spoke
	6	1224	4	2	Spoke hole(s) redefined
	7	1400	2	0	Rim finished
Spoke	1	0	—	27	Start
	2	238	11	16	Finished discussions with Rim
	3	620	4	12	Finished discusions with Hub
	4	725	11	1	Individual CAD work, embody simple spoke
	5	930	–12	13	Event occurs, rework induced in spoke
	6	1012	–4	17	Interupt Hub, results in additional open problems
	7	1145	–3	20	Interupt Rim, further open problems
	8	1224	3	17	Negotiation with Rim finish
	9	1879	9	8	Negotiation with Hub finish
	10	2760	8	0	Spoke finished
Hub	1	0	—	45	Start
	2	238	16	29	Finished discussions with Rim
	3	620	8	21	Finished discusions with Spoke
	4	725	6	15	Individual work: internal hub specification
	5	930	3	12	Started CAD work
	6	1012	–31	43	Spoke interupts, problems re-open
	7	1224	14	29	Rim related interface problems closed
	8	1879	8	21	Negotiation with Spoke finish
	9	1980	6	15	Individual work: internal hub specification
	10	3240	15	0	Hub finished

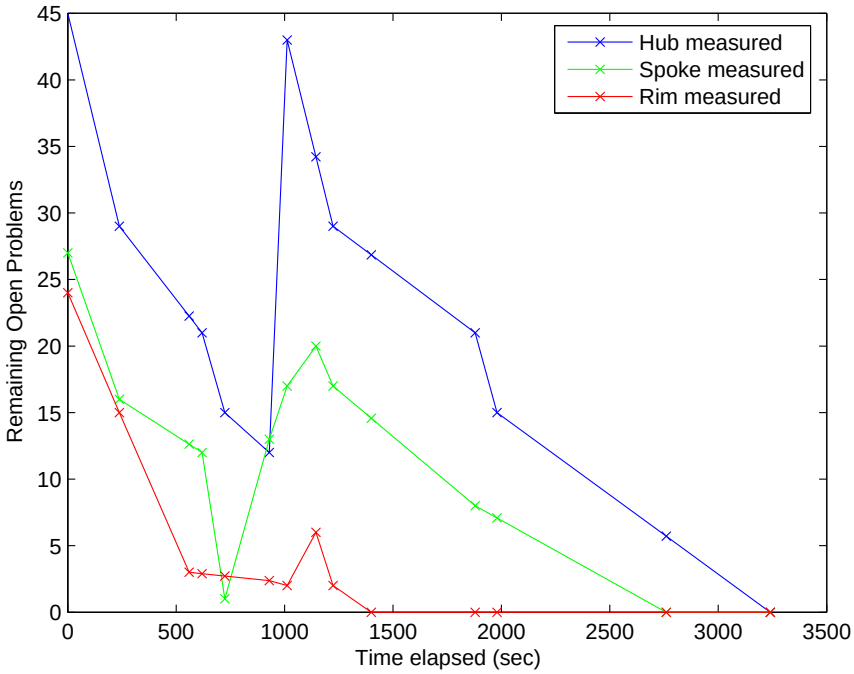


Figure 1: Open problems for design tasks against time elapsed.

simplicity of these components, which enabled (2) the designers to fully discuss and plan their work prior to engaging in their own independent work. This independent working was supported by the evidence collected during the design session. The values of the diagonal were obtained by least squares fitting to the observed data.

Second, there is a simple structure to $\mathbf{B}$, the DTM. The diagonal elements, b_{ii}, are equal to unity. This represents that a component that induces a disturbance will collect the full impact of that disturbance. Further, for non-zero entries, $b_{ij} = b_{ji}^{-1}$. This represents the symmetry that is a result of the pairwise negotiation process. It assumes that any change is negotiated to minimise the pairwise impact, but not necessarily minimise the global impact. Finally, the zero entries represent component pairs that do not directly affect each other. In this case, $b_{31} = b_{13} = 0$ represents the rim and the hub having no directly observable effect on each other. It is worth noting that this is based from the empirical observations, and there is a possibility that these zeros are incorrect.

The values of the DTM matrix were obtained by solving the equation:

$$\mathbf{B}\delta(t = 930) = \mathbf{B} \begin{pmatrix} 0 \\ 16 \\ 0 \end{pmatrix} = \begin{pmatrix} 31 \\ 16 \\ 4 \end{pmatrix} \tag{4}$$

subject to $b_{ii} = 1$, $b_{13} = b_{31} = 0$ and $b_{ij} = b_{ji}^{-1}$ for $ij \neq (1, 3)$. This leaves only b_{21} and b_{23} to be determined. Through matrix algebra: $b_{12} \times 16 = 31$ and $b_{32} \times 16 = 4$, i.e. $b_{12} \approx 2$ and $b_{32} = 0.25$. This completes the DTM matrix.

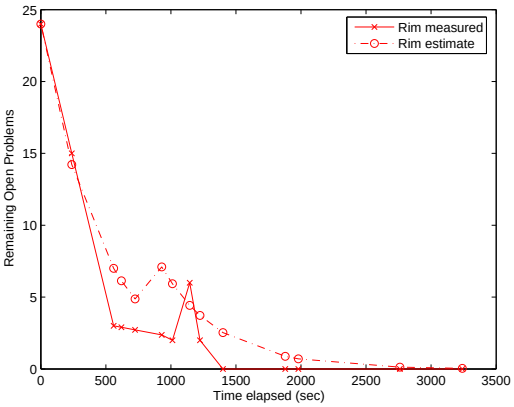
4.5 Model Verification

The model was numerically verified using Equation 2, with matrix values taken from Equation 3, and starting vector of $\mathbf{x}(t = 0) = (45,\ 27,\ 24)^{\mathrm{T}}$. The disturbance was defined as:

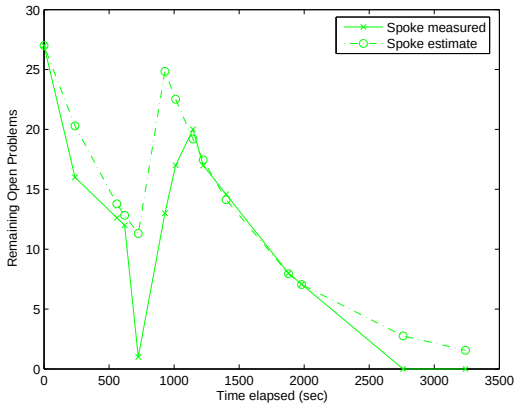
$$\delta(t) = \begin{cases} \mathbf{0} & : \quad (t \neq 930) \\ (0,\ 16,\ 0)^{\mathrm{T}} & : \quad (t = 930) \end{cases} \tag{5}$$

This disturbance function represents that at $t = 930$, a total of 16 problems were re-opened in the spoke component. Figure 2 displays the actual data and the model estimates for the three components.

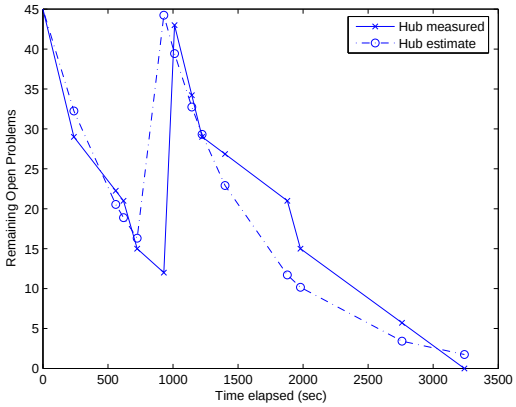
Overall, as can be seen from Figure 2, the WTM/DTM model provides a reasonable estimate of the actual (measured) number of open problems for each of the components. However, there are some noteworthy differences in each of the components. Recall, the disturbance was created in the spoke component. The direct additive means in which the DTM component was included in the estimate model results in the new open problems being propagated to the other components at the same time as the new open problems are incurred in the originating component. This is due to the WTM/DTM model not including any communication latency. However, the overall model fits the data relatively well provided that the communication latency is not great, as is the case in this trial. As can be seen in the Rim (Figure 2(a)) and Hub (Figure 2(c)) graphs, the model estimates the



(a) Rim



(b) Spoke



(c) Hub

Figure 2: Comparison between actual open problems (measured, solid line) and estimated open problems (dashed line) as time elapsed.

additional open problems before they were measured. With regard to the spoke component (Figure 2(b)), there is a considerable discrepancy between the model estimate and the observed values prior to the event. Specifically, the model predicts slower progress than was observed. Post-event, a much closer fit is achieved. A possible reason for this is that the spoke design was significantly more complex, effectively reducing the rate of work possible on the new spoke. A more accurate model in this case would require to update the WTM to reflect this relatively significant design modification.

5 Discussion

The agile design methodology does assume, and requires, that the design project is being undertaken in a virtual enterprise environment. The aim of the agile design methodology is to enable rapid redeployment of design work when required due to an unforseen event occurring. These events are basically design change requirements, arising due either to customer request, internal request resulting from design work, or internal request due to an agent failing to fulfil its task. A classification for the severity of these events was proposed that covered the full spectrum. Within this severity spectrum, it has been argued that the extreme classes are not appropriate for agile design methods. However, events classified away from these extremes provide an opportunity for agile response.

The agile design methodology was tested using a simple trial case. A disturbance to the spoke design was introduced part way through the design process. The resulting propagation was observed and analysed. This disturbance resulted in a significant effect on the hub design. By using some simple analysis on the number of problems that were re-opened as a result of this, a coarse WTM and DTM were estimated (Equation 3).

The DTM provides another means for identifying that the hub design has the most significant effect on the remainder of the design (see Equation 3, value 4 in $\mathbf{B}$, the DTM matrix representing the effect Hub has on the Spoke being the largest by a significant margin). As a result, it can be argued that the hub design agent should lead the redesign negotiation. Minimising the changes to the hub subsequently minimises the changes to the remainder of the design. This results in a more agile design process.

The trial primarily tested the process of interupting other design agents when an unexpected change was introduced. Without this, the affected design agent would have had a considerably greater challenge in providing a component that could still interface with the rest of the assembly, but still responded to the imposed change. In the case of this trial, this would have resulted in an exceptionally long design process for the spoke, as a radically different design would have been required rather than the simple change implemented in this case.

Overall, the estimates and quantitative model fitting were based on a single trial. However, the data collection process and design negotiation processes illustrated that the DTM is a valid extension of the WTM model. Further, it can be seen that from inspection of the DTM matrix, it becomes clear how the design change negotiations should be directed to minimise total design change.

6 Conclusions

This paper introduced a framework for representing and analysing agile design methodology based on concurrent engineering. This illustrated how the design state could be mapped out onto the virtual enterprise network, and provided a means for representing and measuring the effect of unexpected design events occur during the design process. The agile design methodology redistributes the design work after an event occurs according to the severity. The framework would enable virtual enterprises to test various work redistribution policies using the WTM/DTM and Monte Carlo methods to determine the most suitable for different event types.

The next phase of this research will develop methods for computing more accurate numerical values for the WTM and DTM. As a result of this trial, the project data that needs to be collected has been identified. This will provide the high quality quantitative design project data that is required for 'solving' the WTM and DTM for a given product type.

One aspect that has not been considered is the communication latency within a virtual enterprise. It has been assumed that when work is redistributed between agents, this only results in the affected agents 'turning back the clock' on their design process. It was assumed that this resetting of the design process incorporated any latency rather than explicitly accounting for this. However, the trial data clearly illustrated the latency between the initial distrubance and when it finally propagates through to the rim designer. Such latency can be avoided through analysis of the WTM, and research will conducted on developing methods for bringing downstream design teams to the negotiation process more rapidly.

References

[1] J Kusar, J Duhovnik, J Grum, and M Starbek. How to reduce new product development time. *Robotics and Computer-Integrated Manufacturing*, 20:1–15, 2004.

[2] L M Camarinha-Matos and H Afsarmanesh. Elements of VE infrastructure. *Computers in Industry*, 51:139–163, 2003.

[3] G Pahl and W Beitz. *Engineering Design: A Systematic Approach*. Springer-Verlag London, second edition, 1996.

[4] D E Carter and B S Baker. *Concurrent Engineering: The Product Development Environment for the 1990s*. Addison-Wesley, 1991.

[5] N D Armoutis and J Bal. *Building the Knowledge Economy: Issues, Applications, and Case Studies*, chapter E-Business through Competence Profiling, pages 474–482. IOS Press, 2003.

[6] H C W Lau, C W Y Wong, K F Pun, and K S Chin. Virtual agent modelling of an agile supply chain infrastructure. *Management Decision*, 41(7):625–634, 2003.

[7] Z Jiang and R Y K Fung. An adaptive agile manufacturing control infrastructure based on TOPNs-CS modelling. *International Journal of Advanced Manufacturing Technology*, 22:191–215, 2003.

[8] C Terwiesch, C H Loch, and A De Meyer. Exchanging preliminary information in concurrent engineering: Alternative coordination strategies. *Organization Science*, 13(4):402–419, 2002.

[9] B Weber, W Wild, M Lauer, and M Reichert. Improving exception handling by discovering change dependencies in adaptive process management systems. In *Business Process Management Workshops*, volume 4103 of *Lecture Notes in Computer Science*, pages 93–104. Springer, 2006.

[10] S Rinderle and M Reichert. Data-driven process control and exception handling in process management systems. In *Proceedings of Advanced Information Systems Engineering*, volume 4001 of *Lecture Notes in Computer Science*, pages 273–287. Springer, 2006.

[11] S Gonnet, G Henning, and H Leone. A model for capturing and representing the engineering design process. *Expert Systems with Applications*, 33(4):881–902, 2007.

[12] M Heller, D Jager, M Schluter, R Schneider, and B Westfechtel. A management system for dynamic and interorganizational design processes in chemical engineering. *Computers & Chemical Engineering*, 29(1):93–111, 2004.

[13] D V Steward. The Design Structure System: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, 28:71–74, 1981.

[14] R D Smith and S D Eppinger. Identifying controlling features of engineering design iteration. *Management Science*, 43(3):276–293, 1997.

[15] M Carrascosa, S D Eppinger, and D E Whitney. Using the Design Structure Matrix to estimate product development time. In *Proceedings of the DETC'98*, DETC98/DAC-6013, Atlanta, GA, 1998. ASME Design Engineering Technical Conferences.

[16] R P Smith and J Morrow. Product development process modeling. *Design Studies*, 20(3):237–261, 1999.

[17] A Yassine, N Joglekar, D Braha, S Eppinger, and D Whitney. Information hiding in product development: the design churn effect. *Research in Engineering Design*, 14:145–161, 2003.

[18] S G Lee, K L Ong, and L P Khoo. Control and monitoring of concurrent design tasks in a dynamic environment. *Concurrent Engineering: Research and Applications*, 12(1):59–66, 2004.

[19] D N Ford and J D Sterman. The Liar's Club: Concealing rework in concurrent development. *Concurrent Engineering: Research and Applications*, 11(3):211–219, 2003.

[20] P C Matthews, C D W Lomas, N D Armoutis, and P G Maropoulos. Foundations of an agile design methodology. *International Journal of Agile Manufacturing*, 9(1):29–37, 2006.

[21] N D Armoutis, P G Maropoulos, P C Matthews, and C D W Lomas. Establishing agile supply networks through competence profiling. *International Journal of Computer Integrated Manufacturing*, 2008. in press.

[22] N Ye. Information infrastructure of engineering collaboration in a distributed virtual enterprise. *International Journal of Computer Integrated Manufacturing*, 15(3):265–273, 2002.